REMARKS

Claims 1 - 17 remain active in this application. Amendment of claim 11 has been requested above. Support for the amendment of claim 11 is found throughout the application, particularly in Figures 2 through 4 and the description thereof on pages 9 - 11 as well as in claim 1 as originally filed. No new matter has been introduced into the application. The indication of the allowability of the subject matter of claims 5, 9 and 16 is again noted with appreciation.

Claims 1 - 4, 6 - 8 and 10 have again been rejected under 35 U.S.C. §103 as being unpatentable over Bhat in view of Yamada et al. and claims 11 - 15 and 17 have again been rejected under 35 U.S.C. §102 as being anticipated by Sone et al. Both of these grounds of rejection are respectfully traversed for the reasons of record which are hereby fully incorporated by reference and the further remarks provided below.

As previously pointed out, the invention is directed to a technique of suppressing codes which are not needed by a processor when the processor operating in a particular mode or when various peripheral devices may or may not be present. The invention thus can avoid decoding and storage of operations of an application program which will not be needed and thus can improve execution time (e.g. by reducing cache misses or the like since more operations which are likely to be used can be stored locally to the processor) and, more importantly, can save time and power during initialization so that the processor can be re-started, if necessary, in substantially real time.

This meritorious function is achieved without alteration of the application program by inserting execution bits into operation codes of the sequence of operations which form the application program in

accordance with the mode of operation or peripheral devices, if any, which are connected to the processor when the processor is initialized. Then, as the operation codes are decoded and stored for later execution, the state of the execution bit of a current operation being decoded is used to control whether decoding will be continued, in sequence, through the sequence of operation codes or whether one or more subsequent operation codes are to be skipped. This type of control also allows the skipping of operations to be performed in a very simple manner which may be rapidly executed, for example, by simply incrementing the instruction counter. Further, such processing is minimized by placing the execution bit for controlling such processing in an operation which is to be processed to suppress processing of a *subsequent* operation code such that the suppressed operation code *need not be evaluated at all* and is thus fully "skipped". It is respectfully submitted that none of the prior art applied by the Examiner has anything to do with such a function, problem or its solution or lead to any expectation of success in producing the meritorious effects of the invention by the simple expedient provided by the invention, particularly the control of processing of a given operation code using a code in a *preceding* operation code, as clearly and explicitly recited in the claims. Moreover, the Examiner's statements of the above grounds of rejection are entirely silent in regard to such a feature of the invention (as well as glossing over, *inter alia*, the difference between processing of operation codes such as by decoding and storing them and execution of the instructions) and thus do not make a *prima facie* demonstration of anticipation or obviousness of any claim in the application.

For example, Sone et al. is directed to the use of a "slave controller" having a built-in algorithm which

responds to an externally applied command to suspend a series of operations for *executing* the command and to perform a branching operation in accordance with other information in the command in order to perform debugging during *execution* of a program. Sone et al. has nothing to do with processing the operation codes of a program *per se* to control "*processing and storing* a sequence of instructions *to be available for execution*" (claim 11, emphasis added) much less doing so based on a preceding operation code. Therefore, claim 11 clearly distinguishes the processing of operation codes from the execution thereof as well as distinguishing the bypassing of processing from the execution of branching operations or suspension of operations in Sone et al. Accordingly, Sone et al. clearly does not anticipate claim 11 or any claim depending therefrom.

Similarly, as previously pointed out, Bhat is directed to bypassing of processing by a radio cluster server for some classes of communications. While the radio cluster server may be embodied in software, such bypassing of processing, as Bhat is applied by the Examiner, confuses *execution* of operational codes with "processing an application program" in which some instructions "are not used in ones of said plurality of modes" and are "skipped" during such processing responsive to "responsive to detecting a particular state of an execution bit in a preceding instruction" while "decoding remaining operational codes". Further, while Yamada et al. has been cited for teaching the insertion of bits to control *execution* of particular operation codes (again confusing execution of operations with decoding and storing operation codes) the execution bits are contained in the *same* operation codes for which execution is to be controlled. Further, it is evident from even the Abstract of Yamada et al. that the execution control bits are contained in

"the decoded result from thee condition execute decode section" and thus clearly do not control "decoding" of the operation codes, as explicitly recited in claim 1. Therefore, the combination of Bhat and Yamada et al. clearly do not answer the recitations of claim 1 and claims depending therefrom and the Examiner has, in effect, ignored such recitations.

Thus, in summary, it is respectfully submitted that the Examiner has confused the *removal* of unused operations in a program to be decoded and stored for later execution with the *execution* of a program which *contains* operations which are not used; a distinction clearly and explicitly made in the claims. The invention, as claimed, thus supports an acceleration of the decoding and storage of the program by avoiding unnecessary decoding and storage operations as well as accelerating execution of the program by providing more efficient use of storage associated with the processor (e.g. a cache in which cache misses are reduced by storage of more useable operations therein) as well as avoiding the processing of branching during program execution to bypass the unused operations. In this latter regard, it is also respectfully submitted that the Examiner may not ignore or discount remarks regarding functions of the invention which are *supported* by the subject matter of the claims and which are not available from the prior art regardless of whether or not the functions, themselves, are recited in the claims as the Examiner has done. The Examiner's position in this regard is clearly not supported by the authority cited in support thereof. On the contrary, the Examiner has failed to properly consider explicit recitations of the claims which provide clear distinctions from the prior art the Examiner and support the meritorious effects of the invention which the Examiner has confused with the functions of the prior art. In so doing, the Examiner has continued to

fail to make a *prima facie* demonstration of the propriety of either of the asserted grounds of rejection. Therefore, it is clear that these grounds of rejection are in error and, upon reconsideration should be withdrawn.

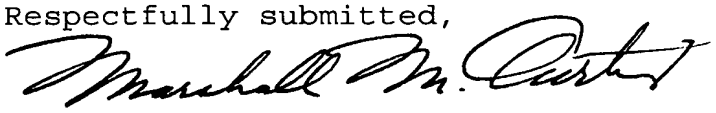
Additionally, it is respectfully submitted that the finality of the present action is premature. While the asserted grounds of rejection have not been changed by the Examiner, it is axiomatic that an action should not be made final when neither that action nor the preceding action contained a *prima facie* demonstration of the propriety of grounds of rejection contained therein. Accordingly, it is respectfully requested that the finality of the present action be withdrawn and the above-requested amendments entered as a matter of right. In any case, entry of the above-requested amendments is respectfully submitted to be well-justified as placing the application in condition for allowance while clearly raising no new issue; the controlling of processing of an operation code in response to content of another operation code having been clearly present in claim 1 and claim 13 as finally rejected. In the alternative, entry is well-justified since the amendments serve to reduce issues and place the application in better form for Appeal.

Since all rejections, objections and requirements contained in the outstanding official action have been fully answered and shown to be in error and/or inapplicable to the present claims, it is respectfully submitted that reconsideration is now in order under the provisions of 37 C.F.R. §1.111(b) and such reconsideration is respectfully requested. Upon reconsideration, it is also respectfully submitted that this application is in condition for allowance and such action is therefore respectfully requested.

If an extension of time is required for this response to be considered as being timely filed, a

conditional petition is hereby made for such extension of time. Please charge any deficiencies in fees and credit any overpayment of fees to Deposit Account No. 09-0456 of International Business Machines Corporation (Burlington).

Respectfully submitted,



Marshall M. Curtis
Reg. No. 33,138

Whitham, Curtis & Christofferson, P. C.
11491 Sunset Hills Road, Suite 340
Reston, Virginia 20190

(703) 787-9400
Customer Number: **30743**